

- 1 -

DATA CONTAINER FOR USER INTERFACE CONTENT DATA

FIELD OF THE INVENTION

5 The present invention relates to a data container and in particular to data containers for use with user interface content data for mobile devices for use with a mobile communications network.

10 BACKGROUND OF THE INVENTION

One of the growth areas for mobile network operators and content providers is the provision of ringtones, wallpapers and other multimedia content for mobile telephones and
15 devices. There is a tension between the needs of mobile network operators and device manufacturers to retain control over some aspects of the device user interfaces for branding purposes and the needs of users to customise and modify the appearance of their devices to suit their own needs. The
20 sophisticated software required to provide the desired flexibility and customisation is also in tension with the limited processing power and data storage capacity of typical mobile devices. The present invention seeks to mitigate these problems.

25

SUMMARY OF THE INVENTION

According to a first aspect of the present invention there is provided a method of provisioning a user interface to a
30 device, the method comprising the steps of a) creating a container, the container comprising: executable code for a user interface; one or more content resources for use in the

user interface; and metadata relating to the or each content resource, the executable code, the or each content resource and the metadata being stored as serialised objects within the container; b) transmitting the container to one or more devices; c) extracting the contents of the container at the or each device; and d) executing the code to generate a user interface for the device.

According to a second aspect of the present invention there is provided a data carrier comprising computer executable code for performing the above described method

According to a third aspect of the present invention there is provided a server for provisioning a user interface to one or more devices, the server comprising: storage means to receive a data container; editing means to enable the data container to be edited, in use the data container comprising executable code for a user interface; one or more content resources for use in the user interface; and metadata relating to the or each content resource, the executable code, the or each content resource and the metadata being stored as serialised objects within the data container; and transmission means for transmitting a data container to one or more devices.

According to a fourth aspect of the present invention there is provided a method of installing a user interface in a device, the method comprising the steps of: a) receiving at a device a container over a communications network, the container comprising: executable code for a user interface; one or more content resources for use in the user interface; and metadata relating to the or each content resource, the executable code, the or each content resource and the

metadata being stored as serialised objects within the container; b) extracting the contents of the container at the device; and c) executing the code to generate a user interface for the device.

5

According to a fifth aspect of the present invention there is provided a data carrier comprising computer executable code for performing the above described method.

10 According to a sixth aspect of the present invention there is provided device comprising a display, a user interface, storage means, processing means and a communication interface, the device being configured, in use, to receive a data container from a communications network via the

15 communications interface; store the data container in the storage means; process the data container using the processing means to extract the contents of the data container, the data container comprising executable code for a user interface; one or more content resources for use in
20 the user interface; and metadata relating to the or each content resource, the executable code, the or each content resource and the metadata being stored as serialised objects within the data container; form a user interface in accordance with the extracted contents of the data container;
25 and display the user interface in the device display.

BRIEF DESCRIPTION OF THE DRAWINGS

30 Figure 1 shows a schematic depiction of a system incorporating the present invention;

- 4 -

Figure 2 depicts in greater detail the structure and operation of server;

Figure 3 shows a schematic depiction of the software 400 for the mobile devices;

5 Figure 4 shows a schematic depiction of the content toolset; and

Figure 5 shows a schematic depiction of a device that comprises a user interface according to an embodiment of the present invention.

10

DETAILED DESCRIPTION OF THE INVENTION

The invention will now be described by way of illustration only and with respect to the accompanying drawings, in which

15 Figure 1 shows a schematic depiction of a system incorporating the present invention. The system comprises server 100, content toolset 200, mobile devices 300, operational support systems (OSSs) 700, content feeds 500 and user interface (UI) sources 600. In use, the server 100

20 communicates content data and UI data to the mobile devices 300, 301, ..., each of which comprise software package 400. The server 100 interfaces with OSSs 700, with the OSSs being those conventionally used to operate mobile networks, for example billing, account management, etc. The server 100

25 further interfaces with the content toolset 200: the content toolset receives data from UI sources 600, 601, ..., and packages the UI data such that the server can transmit the packaged UI data to the software packages 400 comprised within the mobile devices 300. The server receives data from

30 a plurality of content feeds, and this data is processed and packaged such that it can be sent to the software packages 400 or so that the mobile devices 300 can access the data

- 5 -

using the software package 400.

The system can be envisaged as being divided into three separate domains: operator domain 50 comprises the systems and equipment operated by the mobile network operator (MNO); user domain 60 comprises a plurality of mobile devices and third-party domain 70 comprises the content feeds and UI feeds that may be controlled or operated by a number of different entities.

Figure 2 depicts in greater detail the structure and operation of server 100. Server 100 comprises publishing component 110 and content server component 150. Publishing component comprises database 111, import queue 112, content toolset interface 113, user interface 114 & catalogue 115. In operation, the publishing component receives content from the content toolset at the content toolset interface. The content is presented in the form of a parcel 210a, 210b, ..., (see below) comprising one or more Trigs and one or more Triglets. A trig is a user interface for a mobile device, such as a mobile telephone and a triglet is a data file that can be used to extend or alter a trig. If a parcel comprises more than one trig then one of the Trigs may be a master trig from which the other Trigs are derived.

The publishing component user interface 114 can be used to import a parcel into the database 111, and this process causes references to each trig and triglet to be loaded into the import queue 114, which may comprise references to a plurality of parcels 210a, 210b, The contents of the parcel may be examined using the user interface and the contents of the parcel can be passed to the catalogue.

The MNO may have several publishing domains, for example one for each target server in a number of countries or regions. Each domain is treated in isolation from other domains and
5 has its own publishing scheme describing how objects are to be published onto content servers in both live and staging environments. The publishing component GUI provides several different views to each domain, enabling operators to completely manage the publishing of content. The catalogue
10 comprises references to the Trigs stored in the catalogue and the update channels and feed channels used to transfer content to the various domains. For each domain, the operator uses the publishing component GUI to set up the domain structure and allocate trigs from the catalogue to
15 each domain node. To aid the operator in selecting trigs efficiently, a filter is provided in the catalogue so that only relevant items are shown.

A trig may be allocated to several nodes within a domain. In
20 each case the packaging of the trig on the target server may need to be different e.g. a SIS or CAB file, dependent on the handsets that will be accessing the trigs. The packaging can be controlled using the publishing component GUI.

25 The update channels may be referenced by trigs to control the delivery of the content. An update channel comprises a URL which is a link to a resource on the associated domain that comprises a triglet update package. The URL can be polled at predefined intervals and the HTTP GET function used to access
30 the package (it will be readily appreciated that other transport schemes may be used, for example SyncML or SMS or cell broadcast for small updates). The triglet update

package describes how the trig can be modified, e.g. replacing one or more images or text files used by the trig. The publishing component GUI enables an operator to define and control the update channels that exist for a domain, the
5 URLs associated with each triglet on the update channel and the association of triglets with the update channels for a domain. As each triglet is associated with an update channel, an operator may enter the date and time that the update should be published, enabling a schedule to be set.

10

A content feed is similar to an update channel for which the content updates are automatically generated on a regular basis. A content feed is accessed by polling a URL, retrieving the update packet and applying it to the trig.

15 However because of the different nature of manually constructed triglet updates and automatically generated content, update channels and content feeds are managed separately. Again, other transport schemes may be used such as SyncML or OMA-DM (Open Mobile Alliance Device Management).

20

The content server component 150 is a standard implementation of a web server and as such the scaling model is well understood. The capabilities of a server can be rated using a "SPECweb99" number indicating the number of concurrent
25 sessions that the web server can handle under benchmark conditions. Published SPECweb99 numbers range from 404 to 21,000 with typical commercial web servers having SPECweb99 numbers in the order of 5,000. A typical deployment scenario for 1m subscribers with hourly updating content requires a
30 web server with a SPECweb99 rating of only 1,112. A successful deployment will lead to increased service use which can be provided for by enabling additional servers to

create an infrastructure that can be both scalable and highly resilient to failure.

5 A connection may be made to the server from a mobile device via a WAP Gateway. In this case the web server session exists between the WAP gateway and the web server, rather than the mobile phone and web server. When a request is made for a file via the WAP gateway, the session with the web server lasts only as long as it takes to transfer the file from the
10 web server to the WAP gateway - i.e. the session is extremely short since the connection bandwidth will be very high and latency extremely low.

Alternatively a direct connection may be established between
15 the mobile phone and the web server. In this case, the web server will need to keep the session open for as long as it takes to download the data to the phone.

There are two types of content that are delivered by the
20 content server component: trigs, typically of the order of 100KB and regularly updating triglets which are typically of the order of 1KB. The traffic created by trig downloads is very similar to the traffic generated by existing content downloads. And thus the related issues are well understood.
25 Downloads of regular triglet updates are a new feature in an MNO's traffic model but because of the small size of the updates, which typically fit within one data packet, it is possible to show that the traffic can still be handled by typical web servers.

30

Figure 3 shows a schematic depiction of the software 400 for the mobile devices 300, which comprises a mark-up language

renderer 410, update manager 420, network communication agent 425, resource manager 430, virtual file system 435, actor manager 440, a plurality of actors 445a, 445, ..., native UI
5 renderer 450, support manager 460, trig manager 465 and mark-up language parser 470.

It is preferred that the software operates using TrigML, which is an XML application and that mark-up language
10 renderer 410 renders the TrigXML code for display on the mobile device 300. The mark-up language renderer also uses the TrigML Parser to parse TrigML resources, display content on the device screen and controlling the replacement and viewing of content on the handset. The native UI renderer is used to display UI components that can be displayed without
15 the use of TrigML, and for displaying error messages.

The software 400 is provisioned and installed in a device specific manner. For example for a Nokia Series 60 device the software is installed using a SIS file, whereas for a MS
20 Smartphone device the software is installed using a CAB file. Similarly, software upgrades are handled in a device specific manner. The software may be provisioned in a more limited format, as a self-contained application that renders its built in content only: i.e. the software is provisioned with
25 a built-in trig but additional trigs cannot be added later. The supplied trig may be upgraded over the air.

The trig manager 465 presents an interface to the resource manager 430 and the mark-up language renderer. It is
30 responsible for trig management in general. This includes: persisting knowledge of the trig in use, changing the current trig, selection of a trig on start-up, selection of a further

trig as a fall back for a corrupt trig, maintaining the set of installed trigs, identifying where a particular trig is installed to the resource manager and reading the update channel definitions of a trig and configuring the update manager appropriately.

The resource manager provides an abstraction of the persistent store on device, i.e. storing the files as real files, or as records in a database. The resource manager presents a file system interface to the mark-up language renderer and the update manager. It is responsible for handling file path logic, distinguishing between real resource files and actor attributes, mapping trig-relative paths onto absolute paths, interfacing with the trig manager and providing a modification interface to the update manager.

The Update Manager handles the reception and application of Trigs and Triglets. The Update Manager presents an interface to the Renderer and the trig Manager and is responsible for: the initiation of manual updates when instructed to by the Renderer; controlling and implementing the automatic update channel when so configured by the trig manager; indicating the progress of a manual update and recovering an Update following unexpected loss of network connection and/or device power. The update packet format may be defined as a binary serialisation of an XML schema.

XML is a convenient data formatting language that is used to define the update packet format as well as TrigML content. For bandwidth and storage efficiency reasons, text XML is serialised into a binary representation. Both update packets and TrigML fragments are parsed by the same component, the

MARK-UP LANGUAGE PARSER parser. Any further use of XML in the software must use the binary XML encoding and therefore re-use the parser.

5 The Actor Manager 440 looks after the set of actors 445 present in the software. It is used by: the renderer when content is sending events to an actor; actors that want to notify that an attribute value has changed and actors that want to emit an event (see below).

10

The software preferably comprises a multi-threaded application running a minimum of two threads, with more possible depending on how many and what sort of actors are included. The software runs mostly in one thread, referred
15 to as the main thread. The main thread is used to run the renderer which communicates synchronously with other components. Actors always have a synchronous interface to the Renderer. If an actor requires additional threads for its functionality, then it is the responsibility of the Actor to
20 manage the inter-thread communication. It is preferred that a light messaging framework is used to avoid unnecessary code duplication where many actors require inter-thread communication.

25 In addition to the main thread, the update manager runs a network thread. The network thread is used to download update packets and is separate from the main thread to allow the renderer to continue unaffected until the packet has arrived. The Update Manager is responsible for handling
30 inter-thread messaging such that the Update Manager communicates synchronously with the Renderer and Resource

Manager when applying the changes defined in an Update Packet.

The software 400 is provisioned to mobile devices in a device specific method. One or more Trigs can be provisioned as part of the installation, for example, stored as an uncompressed update packet. On start-up, the packet can be expanded and installed to the file-system.

10 The Actors 445 are components that publish attribute values and handle and emit events. Actors communicate with the Renderer synchronously. If an actor needs asynchronous behaviour, then it is the responsibility of the actor to manage and communicate with a thread external to their main
15 thread of the Renderer.

Actor attributes may be read as file references. Attributes are one of four types: a single simple value; a vector of simple values; a single structure of fields, each field
20 having a simple value; or a vector of structures. Attributes may be referenced by an expression using an object.member notation similar to many object-orientated programming languages:

25 <image res="signallevels/{protocol.signalstrength}"/>

When needed as a file, an attribute is accessed via the /attrs folder.

30 <text res="/attr/network/name">

An Actor can be messaged by sending it an event with the <throw> element. Events emitted by actors can be delivered to the content tree as content events: these can be targeted at an element Id or 'top'. The interface to an actor is
5 defined by an Actor Interface Definition file. This is an XML document that defines the attributes, types, fieldnames, events-in and parameters, and events out. The set of actors is configurable at build-time for the software. Appendix A gives an exemplary listing of some actors that may be used,
10 along with the associated functions or variables.

Updates comprise a new trig (a new or replacement UI) or a triglet (a modification to an existing trig) and may be regarded as modifications to the software file-system. The
15 Update Manager to determine what needs changing in the file-system by reading a packet. Update Packets may be downloaded over the air by the software 400 using HTTP, or other suitable transport mechanisms, wrapped in a device-specific package format or pre-provisioned with the installation of
20 the software itself.

There are other failure modes to consider: if an HTTP-GET cannot be initiated, or is met with an HTTP error response code, then this attempt at an Update is abandoned and the
25 retry strategy is used to begin a new update attempt at a later date. Where an HTTP response is interrupted by loss of network signal, any temporary files are deleted and the retry strategy is used to restart the Update attempt at a later date. If an update header indicates that the update payload
30 size may be too great to fit on the device, if the update requires an incompatible version of the software or if the Update already resides on the device then the header data

file is deleted and the Update attempt and any subsequent retries are cancelled.

The content format is common across all platforms
5 implementing the software. The Content Compiler is a content
authoring tool to transform a collection of raw resources
(text TrigML, PNG images, text string definitions) into an
over the air Update Packet that can be written to the file
system of the device.

10

TrigML fragments are files containing text TrigML and
resource references inside these fragments are virtual file
paths. The mapping of these virtual file paths to real file
paths is defined by a TrigDefinition file. This file also
15 defines other properties of the trig. When used for compiling
a triglet, this file also defines how the input
TrigML/PNG/Text resources map onto modifications of the
virtual file-system of a trig.

20 For PNG and Text Resources the trig Definition file points at
a list of real files on the host file-system and the
resources are copied to the outputs.

TrigML can use constant variables instead of attribute
25 values. Constant variables are accessed with the same syntax
as <include> parameters, e.g. \$background_colour. Constants
are treated as global variables in a trig and are defined in
the reserved folder, constants/. The variable definitions
contained in the files in the constants/ folder may be
30 resolved at compile time with direct substitution of their
values. In an alternative embodiment the variable
definitions in constants/ are compiled as global variables

- 15 -

and resolved at content parse time by the software. This allows the trig to be updated by a simple replacement of one or all of its constants files.

5 In order to successfully render the user interface of a mobile device, the mark-up language must have the following qualities: concise page definitions, consistent layout rules, be implementable in a compact renderer, provide multiple layering and arbitrary overlapping content, event model,
10 require the repaint of only the areas of the display that have to change between pages of the UI, include hooks to the platform for reading property values receiving events and sending events, extensible, and be graphically flexible. TrigML provides these features and an overview of the
15 elements and attributes that provide the desired functionality is given in our co-pending application GB0403709.9, filed February 19th 2004.

It is desirable that the cost of re-branding UIs and
20 producing a continual stream of updates is minimal. This is enabled this by providing an efficient flow of information from the creative process through to the transmission of data to users.

25 A container, referred to as a parcel, is used for UIs, UI updates, and templates for 3rd party involvement. Parcels contain all the information necessary for a 3rd party to produce, test and deliver branded UIs and updates.

30 Figure 4 shows a schematic depiction of the content toolset 200, which comprises scripting environment 220, test and simulation environment 230 and maintenance environment 240

- 16 -

The parcel process comprise five processing stages:

- 1) The scripting environment 220 provides the means to design the template for one or more UIs and the update strategy for
5 UIs based on that template.
- 2) The maintenance environment 240 provides for rapid UI and update production in a well-controlled and guided environment that can be outsourced to content providers.
- 3) The maintenance environment 240 'pre-flight' functionality
10 allows the deployment administrator to check and tune the UIs and updates that they receive from 3rd parties.
- 4) The publishing component 110 provides management of UIs and updates at the deployment point, including the staging of new releases.
- 15 5) The publishing component 110 enables the automatic generation of updates from live content feeds.

In a typical project, parcels are created within the scripting environment 220 for: a content provider to create
20 re-branded UIs from a template, incorporating the same 'feel' but a different 'look'; a content provider to create updates from a template, that provide a periodic, or user selected variation to UI content ; or an ad agency to create updates from a template that promote new services on a periodic
25 basis.

For all of these use cases, maintenance environment 240 is used to import the parcel, re-brand and reconfigure the content and create a new parcel for submission to the
30 publishing component 110. In the design of the UI template, the following issues should be considered: which part of the UI can be re-banded; which features of a UI need to be

reconfigured at re-branding or remotely; which part of the UI content may be updated; and if the UI is re-branded then can user select content feeds in use. The scripting environment 220 allows these strategies to be defined, and enables the maintenance environment 240 as the implementer of each instance of each strategy.

The main functions of the scripting environment 220 may comprise:

- 10 • Defines menu structure and page map.
- Defines the framework into which branding content is placed.
- Defines the parts of the UI that are updateable.
- Defines the parts of updates that are replaceable for re-branding.
- 15 • Provides an interactive preview to assist editors
- Provides a graphical code view of each UI layer
- Allows drag and drop of resources into the interactive preview and code view.
- 20 • Exports templates for specific re-branding or update construction tasks
- Simulates UIs and updates on handset simulator.
- Builds UIs and updates for testing on the real device.
- Provides extended debugging tools to aid development.

25 Furthermore, the purpose of the maintenance environment 240 is to provide a designer and administrator's UI for the re-branding and maintenance of skins and updates, with the main functions comprising:

- 30 • Re-branding UI templates
- Populate updates with new content

- 18 -

- Manage UI menu entries via updates
 - Translate UIs and updates for additional languages
 - Purpose strings and content for additional devices
 - Simulation of UIs deployed on handset simulator.
- 5 • Build of UIs and updates for testing on a real device

Rebranded UI

A parcel is generated by the scripting environment 220 which comprises a template UI or update for editing. Once editing
10 is complete the parcel is saved in an 'outbox' ready for despatch to the maintenance environment 240 for publishing to the content server. The following 'parcel' functions are provided. The maintenance environment 240 can be used to edit/replace resources held within the parcel. Parcels can
15 be exported to the simulation environment to test the performance of the UI or UI update on a mobile device.

An explorer is provided for the user to access these categories, with the user being able to change: any UIs or
20 updates marked as visible or the resources within a UI or update that are marked as 'replaceable'. When saved, a thumbnail of the 'visible' object is saved in the parcel, for identification use in the maintenance environment and for other services.

25

A parcel entry may be double clicked to launch an appropriate editor. (for example, an image resource would launch an image editor). All resources may have a text description/note
30 inserted in the maintenance environment and displayed in the appropriate context in the maintenance environment. Lists of menu entries are handled as a special resource type with each

entry presenting its own sub-catalogue of resources (for example - title, help string, image, roll-over image, URL and ringtone preview).

5 Many different UIs can be derived from a common base. Typically the common base would implement most of the interface itself, and Trigs derived from it would implement small variations on it, such as branding. A Triglet can be derived from a Trig, and it can override any of the resources
10 from the parent Trig that it chooses to (optionally it may introduce its own resources). Note that "resources" here also refers to TrigML, so the behaviour and layout of a Trig can be modified by a Triglet just as easily as it replacing a single image or piece of text.

15

A Parcel may comprise one or more base Trigs (i.e. a Trig that is not derived from any other trig), one or more multiple Trigs derived from a base Trig, a plurality of triglets derived from any of the trigs and a plurality of
20 triglets derived from other triglets.

The parcel format is an opaque binary format that stores all this information as serialized objects. The parcel may comprise a number of resources , such as images, text, URLs,
25 update channels, ringtone files, wallpapers, native applications, etc. Each resource contains permission information as to how to view, edit, or delete the resource. Each resource furthermore contains meta information such as documentation and instructions that are relevant to that
30 resource. Each Parcel tool either assumes a relevant role, or requires users to login as a particular role.

The nature of developing trigs is such that a number of people and/or groups of people could be involved in contributing to the final design and implementation of a Trig. Furthermore, the skill sets of these people require
5 that a very simplified and controlled scheme be used to minimize the risk of unwitting damage to the Trig. A typical development workflow for a reasonably complex Trig could comprise:

- 10 • UI Designers create the original UI structure. This design may be built using the maintenance environment to create the first versions of this UI.
- A graphics designer creates the final graphics, and adds them to the design.
- 15 • The areas of the UI dedicated to dynamic content that were identified in the original design need to be fleshed out.
- Graphics for the dynamic update need to be finalised by a graphics designer.
- 20 • Personalisation areas of the UI are then designed and implemented. This might be handled by a number of third-party content providers.

Parcels assist in the workflow described above because they
25 contain the entire project in the single file, and this makes it easy to pass from one member of the team to the next. The Parcel can be re-targeted for the next stage of development by adding comments and instructions on the resources that need to be modified, and even setting the editability of
30 other resources to restrict what can be changed. More complicated workflows can be supported by allowing Parcels to be forked, and separate development to happen in each fork of

the Parcel. Merge tools allow the individual changes to be combined back into a single Parcel. A parcel may be implemented using the pickle module for the Python programming language.

5

The parcels may be used to develop trigs and/or triglets for mobile devices having different capabilities such as display size, RAM capacity. To simplify this, a number of hierarchies may be defined and the data resource or TrigML element classified within the hierarchies. When a trig or triglet is compiled from a parcel, the most appropriate resources or TrigML elements can be selected and compiled for a particular device.

15 Figure 5 shows a schematic depiction of a device 800 that comprises a user interface according to an embodiment of the present invention. The device comprises a display 810 that displays the user interface 815 and user interface means 820, that enable the user to interact with the user interface 815.

20 A processor 830 executes the software that is stored within one or more storage means 840 and there may be provided one or more wireless communication interfaces 850, to enable communication with other devices and/or communication networks. One or more batteries 860 may be received to power
25 the device, which may also comprise interfaces to receive electrical power and/or communication cables.

The nature of these components and interfaces will depend upon the nature of the device. It will be understood that
30 such a user interface can be implemented within a mobile or cellular telephone handset, but it is also applicable to other portable devices such as digital cameras, personal

digital organisers, digital music players, GPS navigators, portable gaming consoles, etc. Furthermore, it is also applicable to other devices that comprise a user interface, such as laptop or desktop computers.

5

The user interface means may comprise a plurality of buttons, such as a numerical or alpha-numerical keyboard, or a touch screen or similar. One or more storage devices may comprise a form of non-volatile memory, such as a memory card, so that
10 the stored data is not lost if power is lost. ROM storage means may be provided to store data which does not need updating or changing. Some RAM may be provided for temporary storage as the faster response times support the caching of frequently accessed data. The device may also accept user
15 removable memory cards and optionally hard disk drives may be used as a storage means. The storage means used will be determined by balancing the different requirements of device size, power consumption, the volume of storage required, etc.

20 Such a device may be implemented in conjunction with virtually any wireless communications network, for example second generation digital mobile telephone networks (i.e. GSM, D-AMPS), so-called 2.5G networks (i.e. GPRS, HSCSD, EDGE), third generation WCDMA or CDMA-2000 networks and
25 improvements to and derivatives of these and similar networks. Within buildings and campuses other technologies such as Bluetooth, IrDa or wireless LANs (whether based on radio or optical systems) may also be used. USB and/or FireWire connectivity may be supplied for data
30 synchronisation with other devices and/or for battery charging.

Computer software for implementing the methods and/or for configuring a device as described above may be provided on data carriers such as floppy disks, CD-ROMS. DVDs, non-volatile memory cards, etc.

5

This application claims the benefit of UK Patent Application number 0403709.9, filed February 19th 2004, the contents of which are incorporated herein by reference.

APPENDIX A

Trigplayer Actor	Attributes	UpdateState	
	Messages	exit	
		predial_mode	on/off
	Events	idle	

Launch Actor	Attributes		
	Messages	browser	url
		SMS	Number
		.	message
		Camera	
		Inbox	
		Profiles	
		missed_calls	
		dialer	number
		...	
		native_app	app_id
			url
	Events		

- 25 -

Install Actor	Attributes		
	Messages	ringtone	resource_path
		wallpaper	resource_path
	Events		

Phone Actor	Attributes	Bluetooth	
		IrDA	
		Call	
		GPRS	
		UnreadSMS	
		UnreadVoiceMail	
		UnreadMsgs	
		BatteryLevel	
		SignalStrength	
	Messages		
	Events	missed_call	
		message_arrived	
		voice_mail_arrived	